AD-A032 689   MICHIGAN UNIV ANN ARBOR DEPT OF ELECTRICAL AND COMPU--ETC F/G 12/1
COMPLEXITY OF VECTORIZED SOLUTION OF TWO-DIMENSIONAL FINITE ELE--ETC(U)
1976   D A CALAHAN   AF-AFOSR-2812-75

UNCLASSIFIED   AFOSR-TR-76-0871   NL

| OF |
AD
A032689

END
DATE
FILMED
1-77

# COMPLEXITY OF VECTORIZED SOLUTION OF TWO-DIMENSIONAL FINITE ELEMENT GRIDS*

D.A. Calahan
Department of Electrical and Computer Engineering
The University of Michigan, Ann Arbor, Michigan 48109

## ABSTRACT

The dissected direct solution of finite-element grids has been shown to be more efficient than band-related methods. In this paper, it is shown that vector processors can be used to solve such grids despite their apparent disassociated structure. Operation counts and vector counts are given as a function of grid size and order of rectangular element.

## INTRODUCTION

The architecture of fourth generation scientific processors (vector processors) places a high premium on regularity of problem data structure. In the class of sparse matrix problems, such structure arises most obviously from the direct solution of finite element (or finite difference) grids resulting from the solution of partial differential equations (Ref.1).

Traditionally, such grids are solved either iteratively or with band-related methods. Although suitable for vector processing, band methods are now known to be computationally inefficient for all but the smallest problems. With benchmarked results (Ref.2) showing vector processing rates 14 to 60 times those of the fastest scalar processors, problem size is growing to the point that failure to use these more elaborate but efficient dissection methods (Ref.3) will result in an order of magnitude loss in computing time, clearly an unacceptable price. On the other hand, dissection methods necessitate processing relatively small matrices in the early stages of solution, and so are unfavorable to vector processing (Ref.3). As the dissected solution process progresses, however, increasingly larger (full) matrices are encountered. It has been an open question whether the small matrices of the initial phases can be compensated by the later

*This research was supported by the Mathematical and Information Sciences Directorate of the Air Force Office of Scientific Research under Grant 75-2812.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| | |
|---|---|
| (19) REPORT DOCUMENTATION PAGE | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM |
| 1. REPORT NUMBER (18) AFOSR - TR - 76 - 0871 ✓ — 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) (6) COMPLEXITY OF VECTORIZED SOLUTION OF TWO-DIMENSIONAL FINITE ELEMENT GRIDS. | 5. TYPE OF REPORT & PERIOD COVERED (9) Interim rept.<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) (10) D. A./Calahan | 8. CONTRACT OR GRANT NUMBER(s) (15) ✓AF-AFOSR ~~75~~-2812-75 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>University of Michigan<br>Dept of Electrical & Computer Engineering ✓<br>Ann Arbor, Michigan 48104 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>61102F (17) (12)<br>(16) 9749-03 14p. |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Air Force Office of Scientific Research (NM)<br>Bolling AFB, DC 20332 | 12. REPORT DATE (11) 1976<br>13. NUMBER OF PAGES 13 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | 15. SECURITY CLASS. (of this report) UNCLASSIFIED<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Vector processing
Parallel processing
Finite elements
Sparse matrix

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The dissected direct solution of finite element grids has been shown to be more efficient than band-related methods. In this paper that vector processors can be used to solve such grids despite their apparent disassociated structure. Operation counts and vector counts are given as a function of grid size and order of rectangular element.

UNCLASSIFIED 407 400

large matrices, so that, overall, the problem appears sufficiently structured to justify vector processing.

Another complication introduced by size is the need for a backing store to keep intermediate results. In reducing the computational complexity by dissection, one risks being overwhelmed by reading and/or writing to memory. To meet space limitations, consideration of this problem has been deleted from this paper but appears in Ref. 3.

In this paper, expressions are developed for operations and vector counts of dissected finite element grids as functions of the grid size, variables/node, and nodes/element. The vectorizability of certain common finite element problems is studied using simple models of current vector processors.

The discussion will assume that the reader is familiar with dissection strategies as described by George (Ref.1), as well as with concepts and terminology associated with solution of problems on vector processors (Ref.2).

## A DISSECTION ALGORITHM FOR ARBITRARY RECTANGULAR ELEMENTS

Consider the sequence of rectangular finite elements illustrated in Fig. 1(a), with m+2 nodes/side and $\ell$ variables/node .* It is assumed that all variables associated with an element are coupled to one another, represented by a full matrix if the variables were ordered consecutively in the grid.

Now consider a square grid of $2^n$-2 elements/side as illustrated in Fig. 1(b). The grid is bisected in both dimensions until only trivial bisections remain. The nodes are eliminated in reverse order, taking care that nodes along the periphery are eliminated as soon as possible. For example, in Fig. 2, a local corner segment of a grid of quadratic elements is illustrated; all nodes marked "1" are eliminated consecutively, followed by nodes marked "2", etc. If each node represents $\ell$ variables, these variables are eliminated consecutively in any order.

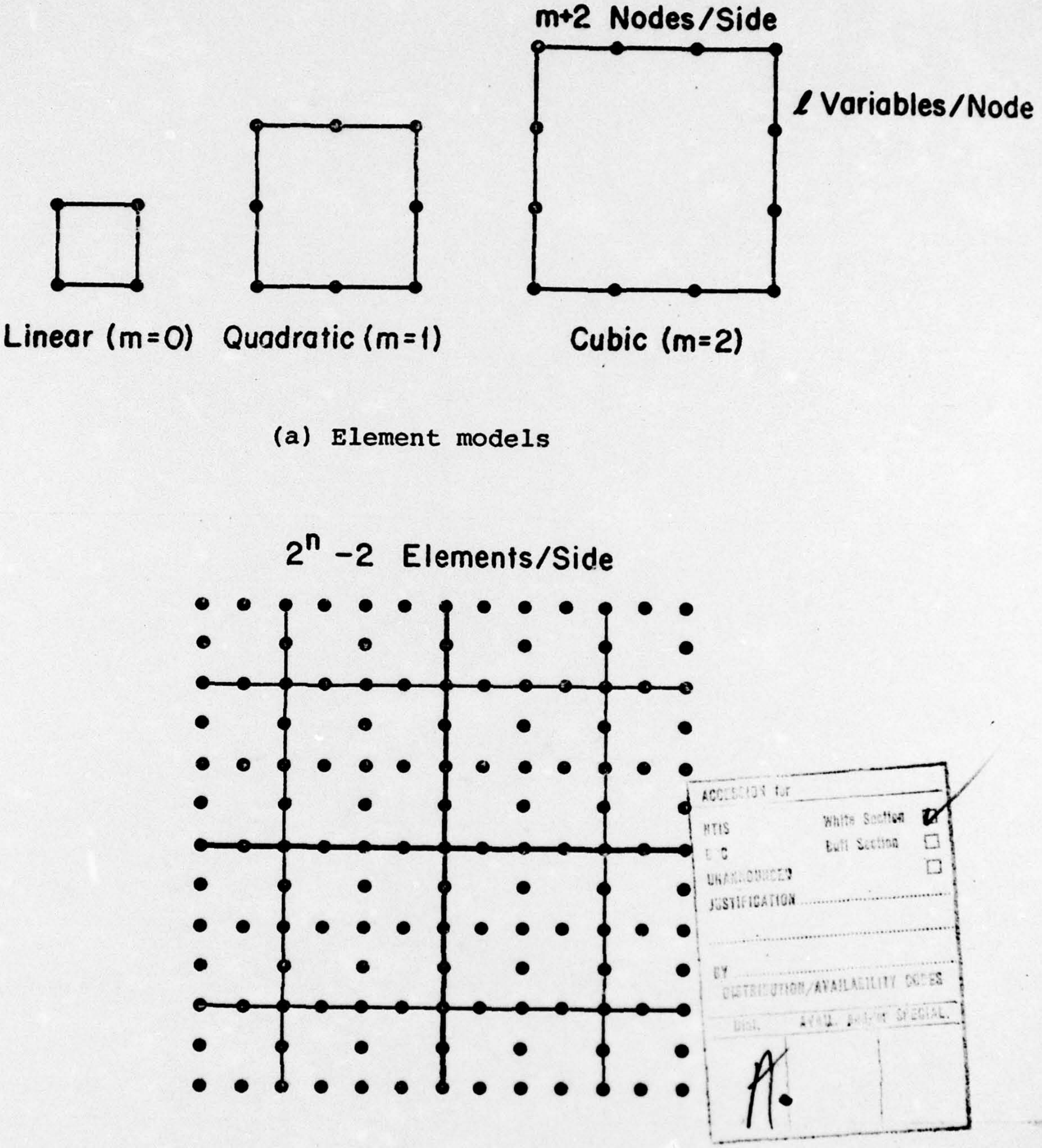
## OPERATION AND VECTOR COUNTS

### Introduction

The characterization of vector operations arising from a dissected finite element solution involves:

(a) identification of vector operations from a matrix structure;

*The terminology "linear," "quadratic," and "cubic" will be used to describe elements with 2, 3, and 4 nodes/side; it should be noted, however, that certain classes of higher-order elements result in more variables/node rather than more nodes/side (e.g., Hermite bicubic).



(a) Element models

(b) Dissected grid example (m=1,n=3)

Figure 1. Finite element and dissected grid examples

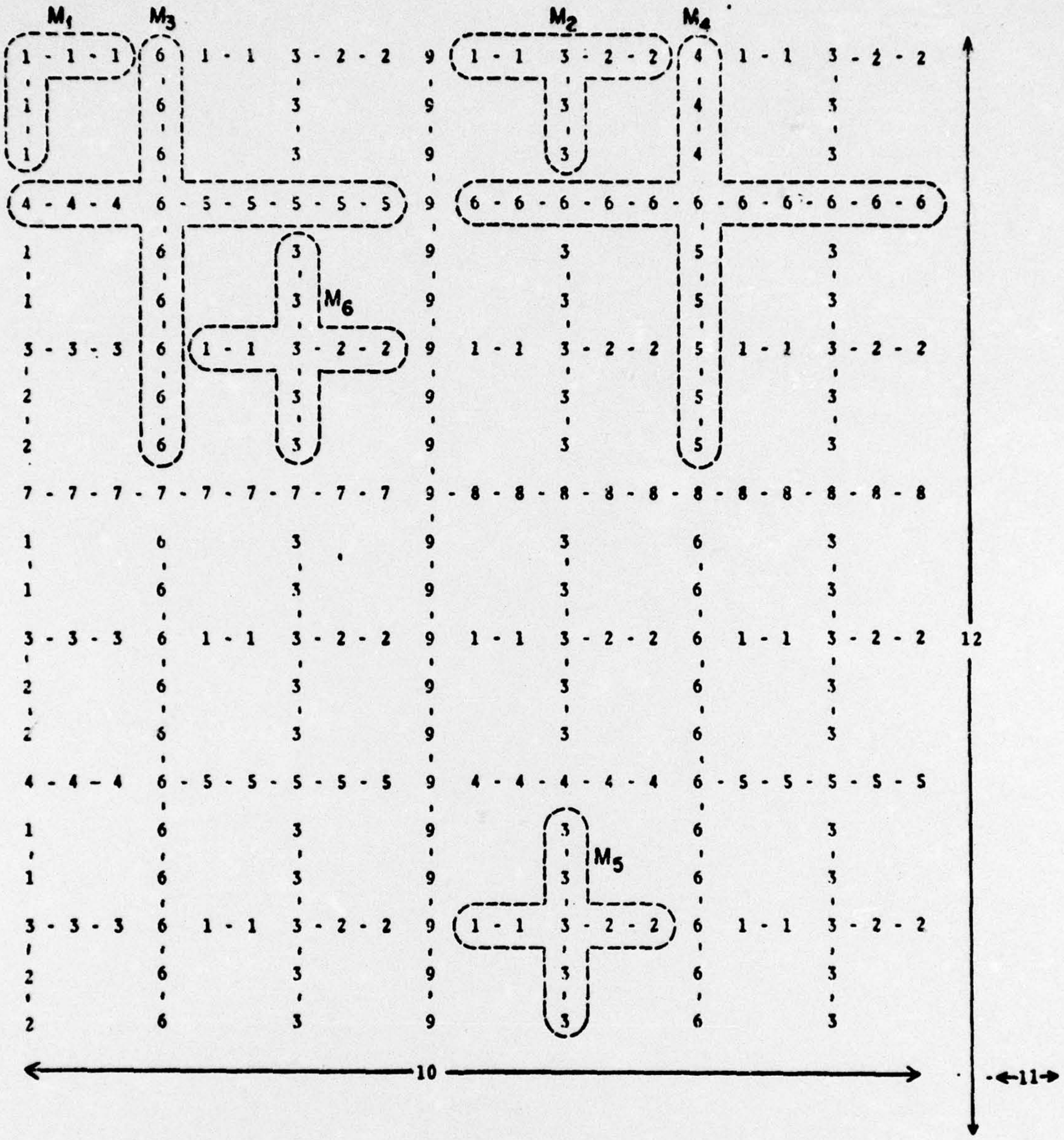


Figure 2. Quadrant of finite element grid. Numbers joined by dashes constitute a single vector.

(b) relation of matrix structure to grid properties; combined with (a), this allows vector operations to be directly related to the grid structure;
(c) developing general expressions for vector counts, with undetermined coefficients;
(d) solving for the coefficients by counting low-order cases.

The arithmetic operation counts are developed similarly to vector counts, using counting formulae developed in Ref. 1. It should be noted, however, that the procedure proposed here does not involve symbolic reduction to arrive at counts as in Ref. 1, but instead solves for the coefficients numerically.

## Simple Vector Counts

Consider the equations

$$\underset{\sim}{A}\underset{\sim}{x} = \underline{b} \tag{1}$$

where $\underset{\sim}{A}$ is a constant $\alpha \times \alpha$ matrix, structurally symmetric but numerically unsymmetric,
$\underline{b}$ is a constant $\alpha \times 1$ matrix,
$\underline{x}$ is an unknown $\alpha \times 1$ matrix.

Then define the LU factorization $\underset{\sim}{L}\underset{\sim}{U} = \underset{\sim}{A}$ where $\underset{\sim}{L}$ and $\underset{\sim}{U}$ are lower and upper triangular matrices, respectively, with unit diagonal entries for $\underset{\sim}{L}$. Then $\underline{x}$ is found by solving $\underset{\sim}{L}\underline{y} = \underline{b}$ and $\underset{\sim}{U}\underline{x} = \underline{y}$.

One manner of viewing this factorization is based on the outer product calculation of Fig. 3a. For counting purposes, a full matrix data structure is assumed; however, simple vectors are recognized and ordered down coluns of $\underset{\sim}{L}$. The r-th simple vector

$$a(I(i(r),j(r)),N(r)) = [a_{1,r}\ a_{2,r}\ \cdots\ a_{N(r),r}]^T$$

begins at position (i(r),j(r)) in the full matrix structure and has length N(r). The index function I(i,j) gives the starting location of a vector in the column-ordered packed matrix obtained by eliminating zero-valued positions. In Fig. 3a, the k-th pivot step is illustrated, and involves operations on t simple vectors in the k-th column of $\underset{\sim}{A}$.

Consider now in detail the process of pivoting at the (k,k) diagonal position. The r-th vector is the first vector identified below the (k,k) diagonal entry. We define

$$s = \sum_{w=1}^{t} N(r + w - 1)$$

where s is the number of non-zero positions in the k-th column. The k-th pivot process begins by a reciprocation of the k-th pivot element and a multiply of the k-th column by the reciprocated pivot (1 vector multiply, s arithmetic operations not including the divide). The entire k-th column is then multiplied by the q-th element of the k-th row (1 vector multiply and s arithmetic operations for each row element). The resultant vector is subtracted from the q-th column; however, the subtraction in general requires t vector operations corresponding to the t vectors in the k-th column which must be scattered in the subtraction from the differently-structured q-th column. This accounts for t vector operations and s arithmetic operations for each of the s components of the k-th row. The total number operations is easily summed at the k-th pivot as

$$VEC(k) = s(t + 1) + 1 \tag{2}$$

$$OPS(k) = s(2s + 1) \tag{3}$$

## High Level Vectors

The above gives a simple vector count (Ref.2) since all vector operations can be performed in a single loop. A processor capable of handling three levels of loop nesting allows the multiplication of any two conformable matrices within one vector operation. If such block structure can be recognized within a problem, then fewer vector operations will be required vis-a-vis a simple vector implementation.

Define a completely coupled set of nodes as a set where (1) all nodes are coupled to one another, and (2) a node not in the set is coupled to one member of the set if and only if it is coupled to all members of the set. This property gives rise to a block matrix structure (Fig. 3b) similar to the single row/column structure of Fig. 3a, so that vectors can be similarly counted.

A summary of the vector operations for an $m \times m$ block coupled to $t$ other blocks is

Pivot broadcast multiplies: $m$

Outer product multiplies: $m$

Subtractions: $(m - 1)\hat{t}^2 + t^2$

where $\hat{t} = t$ if blocks $\underset{\sim}{A}_{11}$ and $\underset{\sim}{A}_{12}$ are adjacent in the matrix

$\hat{t} = t + 1$ if blocks $\underset{\sim}{A}_{11}$ and $\underset{\sim}{A}_{12}$ are nonadjacent.

Thus, at the k-th block elimination, the vector count is

$$\text{VEC}(k) = 2m + (m - 1)\hat{t}^2 + t. \qquad (4)$$

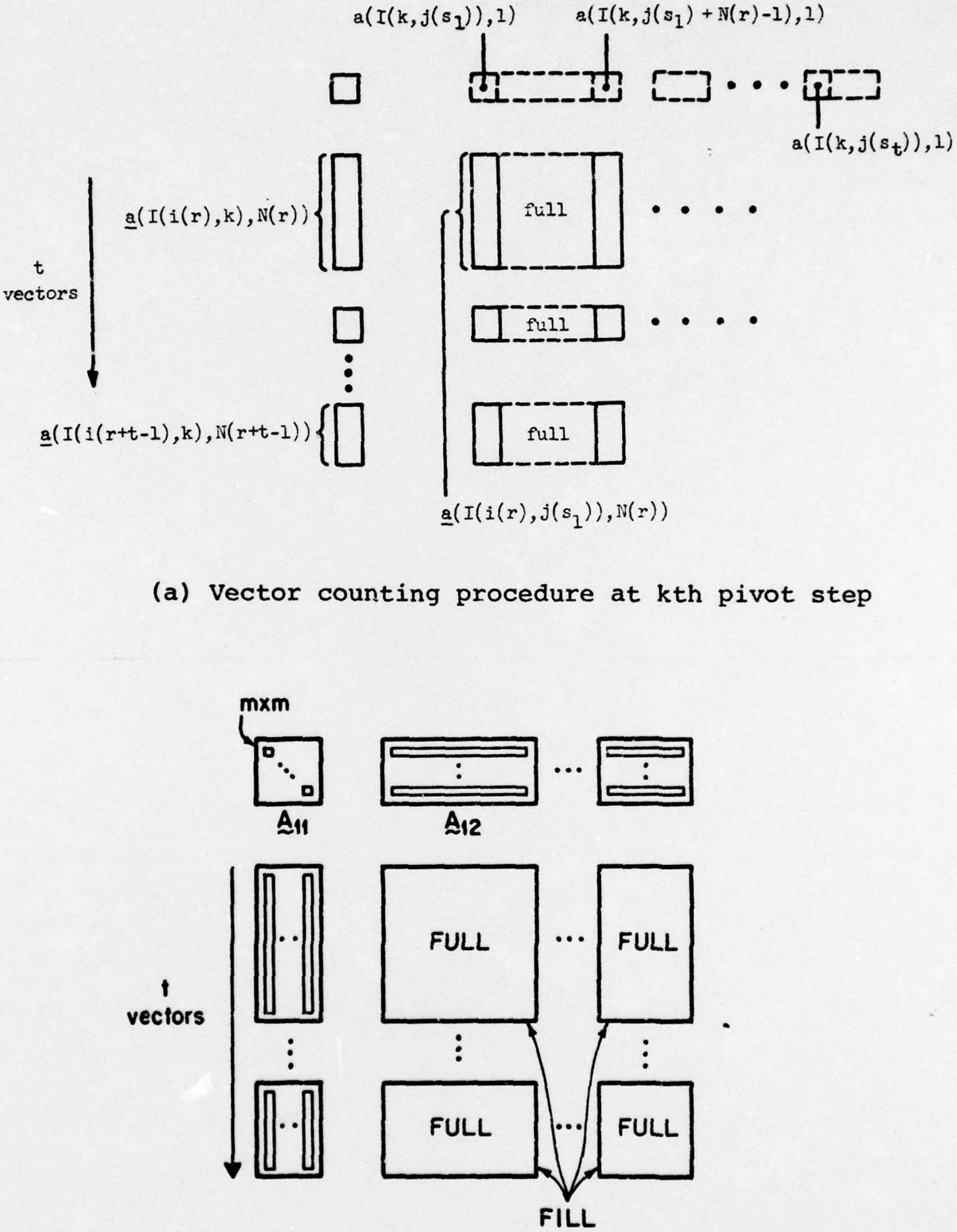
Note that this count does not involve the size of the blocks to which the k-th block is connected.

## Grid Recognition of Vectors

The next step in the determination of an expression for vector count is the ability to recognize vectors in the finite element grid. The general rule for such recognition is the following:

> If at some stage in the elimination process, the node being eliminated (node 1 in Fig. 4) is coupled to a set of consecutively numbered nodes, $n_k, n_{k+1}, \ldots, n_{k+r}$ (e.g., 30-36 in Fig. 4), then consecutive identical operations involving the node set can be performed in a simple vector mode.

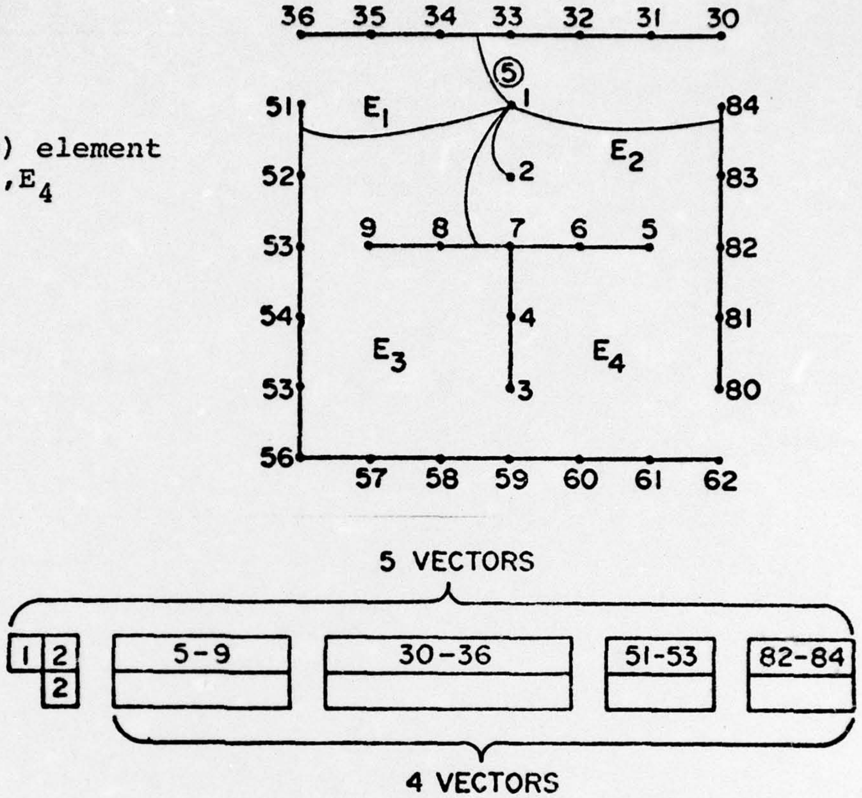
For arbitrary node ordering, such vectors are not readily identified visually; however, the dissection process introduces an ordering along rows and columns of the grid as illustrated in Fig. 4, so that consecutively numbered nodes are usually adjacent in



(a) Vector counting procedure at kth pivot step

(b) Fill pattern of blocked sparse matrix

Figure 3



(a) Four (cubic) element grid $E_1, \ldots, E_4$

(b) Rows of related matrix

Figure 4. Relation of grid ordering to matrix vectors.

the grid structure. Thus, vectors can be identified by a line joining adjacent nodes of a set, as shown in Figure 4. The number of vectors associated with the elimination of selected interior nodes of the example are given in Table 1. It is especially noteworthy that the elimination of the last node of a node set (e.g., node 2 in Fig. 4) involves one fewer vectors than elimination of the previous vector nodes.

Higher lead vectors can be similarly identified by coupling between groups of nodes.

Table 1. Vector Count for Sample Node Elimination in Fig. 4

| Node Eliminated | Node Sets | No. of Vectors |
|---|---|---|
| 1 | 2, 5-9, 30-36, 51-53, 82-84 | 5 |
| 2 | 5-9, 30-36, 51-53, 82-84 | 4 |
| 3 | 4-9, 53-63, 80-82 | 3 |
| 5 | 6-9, 30-36, 51-63, 80-84 | 4 |
| 9 | 30-36, 51-63, 80-84 | 3 |

### Complexity of Vector and Operation Counts

In Ref. 4, it is shown that vector counts for dissected grids are of the general form

$$\mathrm{VEC} = z_o + z_1 n + (z_2 + z_3 n) 2^n + (z_4 + z_5 n) 2^{2n} + (z_6 + z_7 n) 2^{3n} \tag{5}$$

where

$$z_i = w_{i1}\ell + w_{i2}\ell^2 + w_{i3}\ell^2 + (w_{i4}\ell + w_{i5}\ell^2 + w_{i6}\ell^3)m$$
$$+ (w_{i7}\ell + w_{i8}\ell^2 + w_{i9}\ell^3)m^2 + (w_{i10}\ell + w_{i11}\ell^2 + w_{i12}\ell^3)m^3$$

Similar formulae apply to operation counts (Ref. 4), with certain coefficients being zero-valued in each case.

Coefficients are found by counting the vectors and operations for a number of small values of n, $\ell$, and m using (2) and (3) and solving for the $w_{ij}$. For example, 84 grids were simulated to obtain the operation count, and 36 for the simple vector count. The results are summarized in Table 2.

## VECTORIZABILITY OF THE SOLUTION

### Average Vector Length

In Ref. 2 it is shown that the fraction of the computation time devoted to arithmetic operations in a pipelined processor is

$$\eta = \frac{\text{arithmetic time}}{\text{startup time + arithmetic time}}$$
$$= \frac{(T_{op}/T_s)(\sum_{i=1}^{n} \ell_i/n)}{1+(T_{op}/T_s)(\sum_{i=1}^{n} \ell_1/n)} \tag{6}$$

where $T_{op}$ is the arithmetic operation time, $T_s$ the vector startup time, and $\ell_i$ the length of the i-th vector. Although $T_{op}/T_s$ is a machine parameter, the quantity $\sum_{i=1}^{n} \ell_i/n = L_{ave}$ is problem-dependent, and identifiable as the "average vector length." It may be computed from $L_{ave}$ = OPS/VEC, the latter given in (6-10). Its most important property is that when $L_{ave} = T_s/T_{op}$, 1/2 of the total computation time is devoted to vector startup.

### Empirical Study

The average vector lengths resulting from a number of finite element grids have been determined from Eq. (6-10). These range from single variable cubic problems perhaps associated with

TABLE 2. Complexity of Arithmetic and Vector Operators

$$\begin{aligned}
\mathrm{OPS} = {} & [(-\tfrac{1}{6}\ell + 18\tfrac{1}{2}\ell^2 + 126\tfrac{16}{21}\ell^3) + (-\tfrac{2}{3}\ell + 22\ell^2 + 976\tfrac{20}{21}\ell^3)m \\
& + (-11\ell^2 + 1543\tfrac{13}{21}\ell^3)m^2 + 695\tfrac{3}{7}\ell^3 m^3] + 2^n[(\tfrac{1}{3}\ell - 41\ell^2 - 223\tfrac{1}{3}\ell^3) \\
& + (\ell - 84\ell^2 - 410\ell^3)m + (-35\ell^2 - 50\ell^3)m^2 + 134\ell^3 m^3] \\
& + 2^{2n}[(-\tfrac{1}{6}\ell + 23\tfrac{1}{2}\ell^2 + 46\ell^3) + (-\tfrac{1}{3}\ell + 36\ell^2 - 228\tfrac{2}{3}\ell^3)m \\
& + (12\ell^2 - 573\tfrac{1}{3}\ell^3)m^2 - 298\ell^3 m^3] && (6) \\
& + 2^{3n}[39\tfrac{4}{7}\ell^3 + 118\tfrac{5}{7}\ell^3 m + 118\tfrac{5}{7}\ell^3 m^2 + 39\tfrac{4}{7}\ell^3 m^3] && m \geq 0 \\
& + n[(4\ell^2 + 16\ell^3) + (8\ell^2 + 48\ell^3)m + 32\ell^3 m^2] \\
& + n2^n[(-2\ell^2+100\ell^3)+(20\ell^2+436\ell^3)m+(22\ell^2+484\ell^3)m^2+148\ell^3 m^3] \\
& + n2^{2n}[(-7\tfrac{3}{4}\ell^2 - 99\ell^3) + (-15\tfrac{1}{2}\ell^2 - 198\ell^3)m + (-7\tfrac{3}{4}\ell^2 - 99\ell^3)m^2]
\end{aligned}$$

Simple Vectors

$$\begin{aligned}
\mathrm{VEC} = {} & 6\tfrac{1}{6}\ell - 100\tfrac{1}{6}\ell^2 + (6\tfrac{1}{4}\ell + 252\ell^2)2^n + (-7\tfrac{5}{12}\ell - 159\tfrac{1}{21}\ell^2)2^{2n} && (7) \\
& + (-10\ell - 34\ell^2)n + (11\ell + 44\ell^2)n2^n + 44\tfrac{7}{8}\ell^2 n2^{2n} && m = 0
\end{aligned}$$

$$\begin{aligned}
\mathrm{VEC} = {} & [(14\tfrac{1}{6}\ell - 100\tfrac{1}{6}\ell^2) + (-11\tfrac{1}{2}\ell - 162\tfrac{2}{3}\ell^2)m + (5\tfrac{1}{6}\ell^2)m^2] \\
& + 2^n[(10\tfrac{1}{4}\ell + 252\ell^2) + (29\tfrac{1}{4}\ell + 566\tfrac{1}{2}\ell^2)m + (271\tfrac{1}{2}\ell^2)m^2] \\
& + 2^{2n}[(-8\tfrac{11}{12}\ell - 159\tfrac{1}{12}\ell^2) + (-10\tfrac{3}{4}\ell - 255\tfrac{1}{3}\ell^2)m + (-93\tfrac{5}{12}\ell^2)m^2] \\
& + n[(-10\ell - 34\ell^2) + (-20\ell - 80\ell^2)m + (-20\ell^2)m^2] \\
& + n2^n[(11\ell + 44\ell^2) + (11\ell - 66\ell^2)m + (-110\ell^2)m^2] && (8) \\
& + n2^{2n}[44\tfrac{7}{8}\ell^2 + 89\tfrac{3}{4}\ell^2 m + 44\tfrac{7}{8}\ell^2 m^2] && m \geq 1
\end{aligned}$$

High Level Vectors

$$\begin{aligned}
\mathrm{VEC} = {} & -12\ell - 68\ell^2 + (12\ell + 32\tfrac{1}{2}\ell^2)2^n + (-3\tfrac{3}{4}\ell + 25\tfrac{1}{2}\ell^2)2^{2n} + 12\ell^2 n - 54\ell^2 n2^n && (9) \\
& && m = 0
\end{aligned}$$

$$\begin{aligned}
\mathrm{VEC} = {} & [(8\ell - 68\ell^2) + (133\ell^2)m] + 2^n[(16\ell + 32\tfrac{1}{2}\ell^2) - (125\tfrac{1}{2}\ell^2)m] \\
& + 2^{2n}[(-6\ell + 25\tfrac{1}{2}\ell^2) + (49\tfrac{1}{2}\ell^2)m] + n[(12\ell^2) + (56\ell^2)m] \\
& + n2^n[(-54\ell^2) - (54\ell^2)m] \qquad m \geq 1 && (10)
\end{aligned}$$

stress analysis problems to ten-variable linear atmospheric/ocean model elements. The results are plotted in Fig. 5, together with an indication of the solution time (not including input/output) on the Texas Instruments 1-pipe Advanced Scientific Computer. Two cases are shown, the first for simple vectors, the latter for higher level vectors.

For the simple vector case of Fig. 5a, we observe that $L_{ave} \geq 60$ for all matrices requiring 1 minute on the ASC; thus a medium sized transient problem could be solved with the efficiency of $\cong .5$, since $T_S/T_{op} \cong 60$ for the ASC. On the other hand, a 1-hour static problem would result in $L_{ave} \geq 140$ and an efficiency of .74. On the Cray-1, with $T_S/T_{op} = 10$, an efficiency of .5 would be achieved with any problem requiring more than $\cong 1$ second per solution.

It is especially interesting to compare the analytic expressions for arithmetic operations and average vector lengths of banded and dissected solutions. The asymptotic counts are ($n \to \infty$, $\ell \to \infty$, $m = 0$)

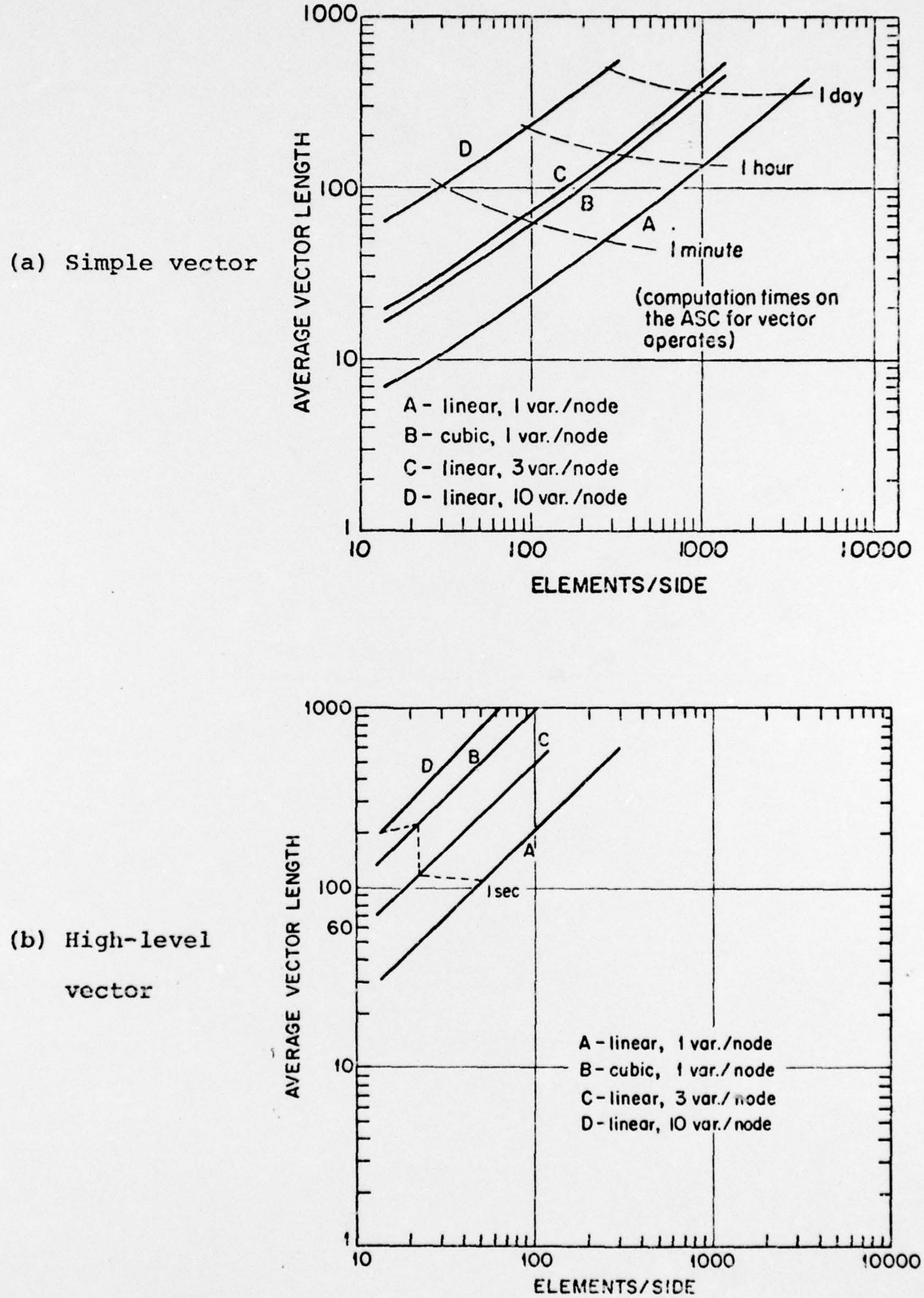
| | | | |
|---|---|---|---|
| banded: | OPS: | $\ell^3 2^{4n}$ | |
| | $L_{ave}$: | $\ell 2^n$ | (simple vector) |
| dissected: | OPS: | $(39\frac{4}{7})\ell^3 2^{3n}$ | |
| | $L_{ave}$:<br>(Ref.3) | $.882\ell 2^n/(n - 3.55)$ | (simple vector) |
| | $L_{ave}$: | $1.55\ell 2^n$ | (high level vector). |

The OPS are less for dissection when $n \geq 6$. The $L_{ave}$ is always larger for a banded solution than a simple vector dissection, but a high level vector dissection is asymptotically more efficient than a simple vector banded solution.

Figure 5b further demonstrates the considerable advantage to be gained in using higher-level vectors. The average vector length, plotted on the same scale as Fig. 5a, is well above 60 for all but the most trivial problem. In this case, vectorization overhead can clearly be ignored.

## CONCLUSIONS

The vector counts given in this paper assumed a data structure that favored highly-coupled systems; e.g., $L_{ave}$ increased with m and $\ell$, which both increase the coupling within an element. Alternatively, the equation decoupling in the early stages of dissection could be exploited as well (Refs. 7,8), but a different data structure would be required. Seemingly, an optimal strategy would involve transitioning from one structure to another as the connectivity grows during solution. This is an interesting but unsolved problem.



(a) Simple vector

(b) High-level vector

Figure 5. Average vector lengths

Apart from the concrete complexity results presented here, the observation that problem dissection does not rule out vectorization has significance to other computational problems as well. For as shown by Rose (Ref.4), dissection is a form of "divide and conquer" algorithm (Ref.5), an efficient general solution strategy which on first inspection does not appear to favor vectorization. The fact that the fill during solution compensates for the initial disassociated structure is perhaps the most interesting of these results.

## REFERENCES

1. George, J.A., "On Block Elimination for Sparse Linear Systems," SIAM J. Numer. Anal., 11, 3, June 1974.

2. Calahan, D.A., "Algorithmic and Architectural Issues Related to Vector Processors," (see Proceedings to this conference).

3. Calahan, D.A., "Complexity of Vectorized Solution of Finite Element Grids," Report 91, Systems Engineering Laboratory, The University of Michigan, November 1975.

4. Rose, D.J., and G.F. Whitten, "A Recessive Analysis of Dissection Strategies," Proc. Symposium on Sparse Matrix Computations, Academic Press, 1976.

5. Aho, A.V., J.E. Hopcraft, and J.D. Ullman, The Design and Analysis of Computer Algorithms, Addison Wesley, 1974.

6. Knight, J.C., et al., "System Balance Analysis for Vector Computers," Report 75-6, ICASE, NASA Langley Research Center, Hampton, Virginia, March 1975.

7. Calahan, D.A., "Parallel Solution of Sparse Simultaneous Linear Equations," Proc. Eleventh Fullerton Conf. on Circuits and Systems, University of Illinois, 729-735, October 1973.

8. Wing, O., and J. Huang, "An Experiment in Parallel Processing of Gaussian Elimination of a Sparse Matrix," Proc. International Symp. on Circuits and Systems, Munich, Germany, April 1976.